Robotics and Autonomous Systems 58 (2010) 140–148

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Bridging the gap between feature- and grid-based SLAM

Kai M. Wurm *, Cyrill Stachniss, Giorgio Grisetti

University of Freiburg, Department of Computer Science, Georges-Köhler-Allee 79, 79110 Freiburg, Germany

ARTICLE INFO

Article history:
Available online 22 September 2009

Keywords:
SLAM
Features
Grid maps
Learning
Dual representation

ABSTRACT

One important design decision for the development of autonomously navigating mobile robots is the choice of the representation of the environment. This includes the question of which type of features should be used, or whether a dense representation such as occupancy grid maps is more appropriate. In this paper, we present an approach which performs SLAM using multiple representations of the environment simultaneously. It uses reinforcement to learn when to switch to an alternative representation method depending on the current observation. This allows the robot to update its pose and map estimate based on the representation that models the surrounding of the robot in the best way. The approach has been implemented on a real robot and evaluated in scenarios, in which a robot has to navigate in- and outdoors and therefore switches between a landmark-based representation and a dense grid map. In practical experiments, we demonstrate that our approach allows a robot to robustly map environments which cannot be adequately modeled by either of the individual representations.

1. Introduction

Building maps is one of the fundamental tasks of mobile robots. In the literature, the mobile robot mapping problem is often referred to as the *simultaneous localization and mapping (SLAM)* problem. It is considered to be a complex problem, because for localization a robot needs a consistent map of the environment, and for acquiring a map, a robot requires a good estimate of its location. This mutual dependency between the estimates about the pose of the robot and the map of the environment makes the SLAM problem hard and involves searching for a solution in a high-dimensional space.

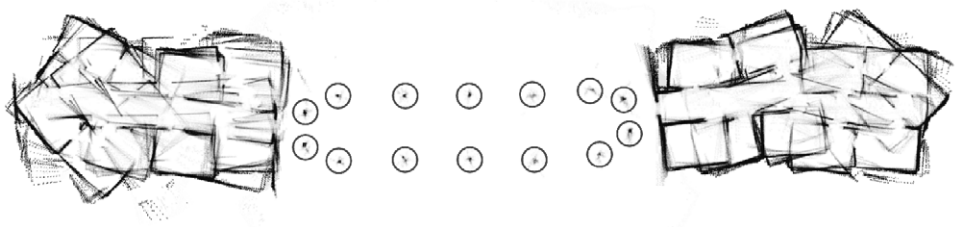
A large variety of different estimation techniques has been proposed to address the SLAM problem. Extended Kalman filters, sparse extended information filters, maximum likelihood methods, particle filters, and several other techniques have been applied to estimate the trajectory of the robot as well as a map of the environment. Most approaches to mapping use a single scheme for representing the environment. Among the most popular ones are feature-based models such as sets of landmarks or dense representations such as occupancy grids. In a practical robotic application, the decision of which model to use is largely influenced by the type of the environment the robot is deployed in. In large open spaces with predefined landmarks, for example, feature-based approaches often are preferred, whereas occupancy grid maps have widely been used in unstructured environments. In real world scenarios, however, one generally cannot assume that the environment is uniformly covered by specific features. Consider, for example, a surveillance system which can operate both inside of buildings and outside on parking spaces or large outdoor storage areas. Such a system has to be capable of dynamically choosing the best representation in each area to maximize its robustness.

The contribution of this paper is a novel approach which allows a mobile robot to utilize different representations of the environment. In the example of a combination of feature-based models with occupancy grid maps, we describe how a robot can perform the mapping process using both types of representation. It applies reinforcement learning to select the representation that is best suited to model the area surrounding the robot based on the current sensor observations and the state of the filter. We apply the approach in the context of a Rao–Blackwellized particle filter to maintain the joint posterior about the trajectory of the robot and the map of the environment.

As we will demonstrate in the experiments, our approach outperforms pure grid and pure feature-based approaches. Furthermore, our approach allows for the modeling of heterogeneous environments which cannot be adequately represented by either of the single representations. A motivating example is shown in Fig. 1. Here, the environment consists of outdoor and indoor parts. A feature-based representation is well suited to model the outdoor part (Fig. 1(a)) but cannot be used to correct odometry errors inside the buildings due to the lack of relevant features. A grid-based representation, in contrast, leads to false matches in the outdoor parts due to the sparsity of non max-range measurements there but accurately represents the inside of the buildings (see Fig. 1(b)). Our system combines the advantages of both representations to generate a consistent map (Fig. 1(c)).

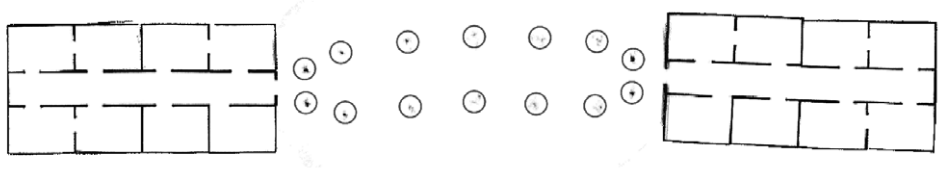
* Corresponding author.
E-mail address: wurm@informatik.uni-freiburg.de (K.M. Wurm).

0921-8890
doi:10.1016/j.robot.2009.09.009



(a) Feature-based mapping system (no features inside the buildings). Features are illustrated by circles.

(b) Grid-based mapping system (few structural information outside).



(c) Combining features and grid maps.

Fig. 1. When mapping environments that contain large open spaces with few landmarks as well as dense structures, a combination of feature maps and grids maps outperforms the individual techniques.

This paper is organized as follows. After a discussion of related work, we briefly introduce the SLAM approach utilized in this paper, namely Rao–Blackwellized particle filters, in Section 3. Whereas Section 4 presents our approach for mapping with a dual representation of the environment, Section 5 explains our model selection technique based on reinforcement learning. Finally, we present experimental results obtained in simulation and on real robots in Section 6.

2. Related work

Mapping techniques for mobile robots can be roughly classified according to the map representation and the underlying estimation technique. One popular map representation is the occupancy grid [1]. Whereas such grid-based approaches are computationally expensive and typically require a huge amount of memory, they are able to represent arbitrary objects. It should be noted that to correct the robot pose estimate a certain amount of obstacles in the range of the robot's sensor is needed. This can be a problem if the range of the sensor is short as is the case with small scale laser scanners or if the environment is a large open area.

Feature-based representations are attractive because of their compactness. This is a clear advantage in terms of memory consumption and processing speed. However, such systems rely on predefined feature extractors, which assume that some structures in the environments are known in advance. This clearly limits the field of action of a robot.

The model of the environment and the applied state estimation technique are often coupled. One of the most popular approaches are extended Kalman filters (EKFs) in combination with predefined landmarks. The effectiveness of the EKF approaches results from the fact that they estimate a fully correlated posterior about landmark maps and robot poses [2,3]. Their weakness lies in the strong assumptions that have to be made on both the robot motion model and the sensor noise. Moreover, the landmarks are assumed to be uniquely identifiable. There exist techniques [4] to deal with unknown data association in the SLAM context, however, if these assumptions are violated, the filter is likely to diverge [5–7].

Thrun et al. [8] proposed a method that uses the inverse of the covariance matrix. The advantage of the sparse extended information filters (SEIFs) is that they make use of the approximative sparsity of the information matrix and in this way can perform predictions and updates in constant time. Eustice et al. [9] presented a technique to make use of exactly sparse information matrices in a delayed-state framework.

In a work by Murphy, Doucet, and colleagues [10,11], Rao–Blackwellized particle filters (RBPF) have been introduced as an effective means to solve the SLAM problem. Each particle in a RBPF represents a possible trajectory of the robot and a map of the environment. The framework has been subsequently extended by Montemerlo et al. [12,13] for approaching the SLAM problem with landmark maps. To learn accurate grid maps, RBPFs have been used by Eliazar and Parr [14] and Hähnel et al. [15]. Whereas the first work describes an efficient map representation, the second presents an improved motion model that reduces the number of required particles. The work of Grisetti et al. [16] describes an improved variant of the algorithm proposed by Hähnel et al. [15] combined with the ideas of FastSLAM2 [12]. Instead of using a fixed proposal distribution, the algorithm computes an improved Gaussian proposal distribution on a per-particle basis on the

fly. A further extension of this method which overcomes the limitation of the Gaussian assumption has recently been presented by Stachniss et al. [17]. Additional improvements concerning both runtime and memory requirements have been achieved by Grisetti et al. [18] by reusing already computed proposal distributions.

So far, there exist only very few methods that try to combine feature-based models with grid maps. One is the hybrid metric map (HYMM) approach [19]. It estimates the location of features and performs a triangulation between them. In this triangulation, a so called dense map is maintained which can be transformed according to the update of the corresponding landmarks. This allows the robot to obtain a dense map by using a feature-based mapping approach. However, it is still required that the robot is able to reliably extract landmarks.

A hybrid map is also used in [20]. Sim et al. propose a vision-based SLAM system which extracts 3D point landmarks from stereo camera images. In addition to the map of landmarks, an occupancy grid map is constructed which is used for safe navigation of the robot. In contrast to the approach described in this paper, the SLAM-system is only using the feature map for pose estimation, while the grid map is used for path planning in an exploration task. A similar approach is described by Makarenko et al. [21]. Here, an decision-theoretic exploration algorithm is described which uses a feature map for SLAM and maintains a grid map to determine known and unknown regions of the environment. However, the grid map is not used to correct the estimate of the robot's pose. Another combination of grid and feature maps has been proposed by Ho and Newman [22]. They use grid maps and visual features in a SLAM system. While the grid map generated from laser scans is used for pose estimation, visual features are used to improve the detection of loop closures.

3. Mapping with Rao–Blackwellized particle filters

According to Murphy [11], the key idea of the Rao–Blackwellized particle filter for SLAM is to estimate the joint posterior $p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1})$ about the map m and the trajectory $x_{1:t} = x_1, \ldots, x_t$ of the robot. This estimation is performed given the observations $z_{1:t} = z_1, \ldots, z_t$ and the odometry measurements $u_{1:t-1} = u_1, \ldots, u_{t-1}$ obtained by the mobile robot. The Rao–Blackwellized particle filter for SLAM makes use of the following factorization

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1}) = p(m \mid x_{1:t}, z_{1:t}) \cdot p(x_{1:t} \mid z_{1:t}, u_{1:t-1}). \quad (1)$$

This factorization allows us to first estimate only the trajectory of the robot and then to compute the map given that trajectory. This technique is often referred to as Rao–Blackwellization.

Typically, Eq. (1) can be calculated efficiently since the posterior about maps $p(m \mid x_{1:t}, z_{1:t})$ can be computed analytically using "mapping with known poses" [1] since $x_{1:t}$ and $z_{1:t}$ are known.

To estimate the posterior $p(x_{1:t} \mid z_{1:t}, u_{1:t-1})$ about the potential trajectories, one can apply a particle filter. Each particle represents a potential trajectory of the robot. Furthermore, an individual map is associated with each sample. The maps are built from the observations and the trajectory hypothesis represented by the corresponding particle.

This framework allows a robot to learn models of the environment and estimate its trajectory, but it leaves open how the environment is represented. So far, this approach has been applied using feature-based models [12,13] or grid maps [14,16,15,11]. Each representation has its advantages and one typically needs some prior information about the environment to select the appropriate model. In this paper, we combine both types of maps to represent the environment. This allows us to combine the advantages of both worlds. Depending on the most recent observation, the robot selects that model which is likely to be the best model in the current situation. In case the environment suggests the use of one single model, the result is the same as using the original approach.

4. Dual model of the environment

Our mapping system applies such a Rao–Blackwellized particle filter to maintain the joint posterior about the trajectory of the robot and the map of the environment. In contrast to previous algorithms, each particle carries a grid map as well as a map of features. The key idea is to maintain both representations simultaneously and to select in each step the model that is best suited to update the pose and map estimate of the robot. Our approach is independent of the actual features that are used. In our current system, we use a laser range finder and extract clusters of beam end points which are surrounded by free space. In this way, we obtain features from trees, street lamps, etc. Note that other feature detectors can be transparently integrated into our approach. The detector itself is completely transparent to the algorithm.

In each step, our algorithm considers the current estimate as well as the current sensor and odometry observation to select either the grid or the feature model to perform the next update step. This decision affects the proposal distribution in the particle filter used for mapping. The proposal distribution is used to obtain the next generation of particles as well as to compute the importance weights of the samples.

In the remainder of this section, we first introduce the characteristics of our particle filter. We then explain in the subsequent section how to actually select the model for the current step.

If the grid map is to be used, we draw the new particle poses from an improved proposal distribution as introduced by Grisetti et al. [16]. This proposal performs scan-matching on a per particle basis and then approximates the likelihood function by a Gaussian. This technique has been shown to yield accurate grid maps of the environment, given that there is enough structure to perform scan-matching for an initial estimate.

When using feature maps, we apply the proposal distribution as done by Montemerlo et al. [13] in the FastSLAM algorithm. For each particle $s_{t-1}^{(i)}$ in the current particle set a new hypothesis of the robot's pose is generated by sampling from the probabilistic motion model:

$$s_t^{(i)} \sim p(s_t \mid u_t, s_{t-1}^{(i)}). \quad (2)$$

After the proposal is used to obtain the next generation of samples, the importance weights are computed according to Grisetti et al. [16] and Montemerlo et al. [13] respectively. Note that we compute for each sample i two weights $w_g^{(i)}$ (based on the grid map) and $w_f^{(i)}$ (based on the feature map). For resampling, one weight is required but we need both values in our decision process as explained in the following Section 5.

To carry out the resampling step, we apply the adaptive resampling strategy originally proposed by Doucet [23]. It computes the so-called effective sample size or effective number of particles (N_{eff}) to decide whether to resample or not. This is done based on the weights resulting from the proposal used to obtain this generation of samples.

5. Model selection

Probably the most important aspect of our proposed algorithm is to decide which representation to choose given the current sensor readings and the filter. In the following, we describe different strategies we investigated and which are evaluated in the experimental section of this paper.

5.1. Observation likelihood criterion

A mapping approach that relies on scan-matching is most likely to fail if laser readings cannot be aligned to the map generated

so far. For example, this will probably be the case in large open space with sparse observations. In such a situation it is often better to use a pre-defined feature extractor (in case there are feature) to estimate the pose of the robot.

A measure that can be used to detect such a situation is the observation likelihood that scan-matching seeks to maximize

$$l(z_t, x_t, m_{g,t}) = \max_{x_t} p(z_t \mid x_t, m_{g,t}). \tag{3}$$

To point-wise evaluate the observation likelihood of a laser observation, we use the so called "beam endpoint model" [24]. In this model, the individual beams within a scan are considered to be independent. The likelihood of a beam is computed based on the distance between the endpoint of the beam and the closest obstacle in the map from that point.

Calculating the average likelihood for all particles results in a value that can be used as a heuristic to decide which map representation to use in a given situation:

$$\bar{l} = \frac{1}{N} \sum_i l(z_t, x_t^{(i)}, m_{g,t}^{(i)}). \tag{4}$$

A *heuristic* for selecting the feature-based representation instead of the grid map can be obtained based on a threshold ($\bar{l} \leq c_1$).

However, care has to be taken when choosing c_1. If this threshold is not chosen optimally the feature map might be used even if it offers no advantage over the grid map. This will increase the likelihood of a poor state estimate and therefore of inconsistencies in the map.

5.2. N_{eff} criterion

As described above, each particle i carries two weights $w_g^{(i)}$ and $w_f^{(i)}$, one for the grid-map and one for the feature-map. These weights can be seen as an indicator of how well a particle explains the data and therefore can be also used as a heuristic for model selection. Since the weights of a particle are based on different types of measurement, they cannot be compared directly. What can be compared, however, is the weight distribution over the filter.

One way to measure this difference in the individual weights is to compute the variance of the weights. Intuitively a set of weights with low variance does not strongly favor any of the hypothesis represented by the particles, while a high variance indicates that some hypotheses are more likely than others.

This suggests that a strategy based on the N_{eff} value, which is strongly related to the variance of the weights, can be a reasonable heuristic. N_{eff} is computed for both sets of weights as

$$N_{\text{eff}}^g = \frac{1}{\sum_{i=1}^{N} (w_g^{(i)})^2} \quad \text{and} \quad N_{\text{eff}}^f = \frac{1}{\sum_{i=1}^{N} (w_f^{(i)})^2}. \tag{5}$$

It can be easily seen, that a higher variance in the weights yields a lower N_{eff} value. Assuming that a set of particles with a higher variance in the weights is usually more discriminative, it seems reasonable to switch to the feature-based model whenever $N_{\text{eff}}^f < N_{\text{eff}}^g$.

In our experiments, this heuristic generally led to good results. Nevertheless, there are two aspects which have to be considered.

Firstly the variance in particles weights usually does not change abruptly but gradually. For this reason, the N_{eff} criterion might fail to indicate the optimal point in time to switch the actively used representation. This will most notably happen at junctions between areas where one is best modeled using grid maps and the other is best modeled using feature maps. Note that such a behavior can also be advantageous, for example in case of false feature detections.

Algorithm 1 The SARSA Algorithm

```
Initialize Q(s, a) arbitrarily
for all episodes do
  initialize s
  choose a from s using policy derived from Q
  repeat
    take action a, observe r, s'
    choose a' from s' using policy derived from Q
    Q(s, a) = Q(s, a) + α[r + γQ(s', a') − Q(s, a)]
    s = s'; a = a'
  until s is a terminal state
end for
```

A second problem arises from the fact that frequent resampling in a particle filter can lead to particle depletion [23]. Since our implementation uses adaptive resampling based on the N_{eff} value, choosing the representation with the lower N_{eff} will in general also lead to more frequent resampling actions.

5.3. Reinforcement learning for model selection

Both approaches described above are clearly heuristics. In this section, we describe how to use reinforcement learning to combine the strengths of both heuristics while avoiding their pitfalls. The basic idea of reinforcement learning is to find a mapping from states S to actions A which maximizes a numerical reward signal r (see [25] for an introduction). Such a mapping is called a policy and can be learned by interacting with the environment. Inspired by the human learning method of trial and error, this class of learning algorithms performs a series of actions and analyzes the obtained reward.

There exist a number of algorithms for reinforcement learning. Depending on the prior knowledge an agent has about its environment some approaches may be more appropriate than others. For example, if it can be modeled as an Markov decision process, techniques such as policy iteration can be utilized. In case no model of the environment is available, Monte Carlo methods or Temporal-Difference Learning (TD learning) can be applied.

For our approach, we use the SARSA algorithm [25] which is a popular algorithm among the TD methods and does not require a model of the environment. It learns an action-value function $Q(s, a)$ which assigns a value to state-action pairs. Those values can then be used to generate a policy (e.g., choose the action that has the highest value in a given state). The basic steps are given in Algorithm 1.

To apply this method to our model selection problem, we have to define the states S, the actions A, and the reward $r : S \to R$. Defining the actions is straight forward as $A = \{a_g, a_f\}$, where a_g defines the use of the grid map and a_f the use of the feature map.

The state set has to be defined in a way that it represents all necessary information about the sensor input and the filter to make a decision. To achieve this, our state consists of the average scan matching likelihood $\bar{l}$, a boolean variable given by $N_{\text{eff}}^f < N_{\text{eff}}^g$, and a boolean variable indicating if a known feature has currently been detected or not. This results in

$$S := \{\bar{l}\} \times \{1_{N_{\text{eff}}^f < N_{\text{eff}}^g}\} \times \{1_{\text{feature detected}}\}. \tag{6}$$

The value of $\bar{l}$ is divided into (here seven) discrete intervals (0.0–0.15, 0.16–0.3, 0.31–0.45, 0.46–0.6, 0.61–0.75, 0.76–0.9, 0.91–1.0), resulting in $7 \times 2 \times 2 = 28$ states. It is important to keep the number of states small since learning the policy otherwise may require too many computational resources, even as a preprocessing step which needs to be executed only once.

The policy is learned on simulated data where the true robot pose x_t^* is available in every time step t. We use the weighted

average deviation from the true pose to define our reward-function. To avoid a punishment that result from wrong decisions in the past (e.g., a wrong rotation), we only use the deviation accumulated since the last evaluation step $t-1$:

$$r(s_t) = r(s_{t-1}) - \sum_{i=1}^{N} w_t^{(i)} \|x_t^{(i)} - x_t^*\|. \quad (7)$$

Deviations from the simulated path result in negative rewards. As mentioned in Section 5, each particle stores two weights. For calculating the weighted average, we use $w_g^{(i)}$ if the last action taken was a_g and $w_f^{(i)}$ if a_f was taken.

The environment for learning consists of building-like structures with hallways and an outdoor part that models a set of trees. We recorded a simulated path and executed 1000 runs of the learning algorithm. During learning, we us an ε-greedy policy. In state s, a greedy policy chooses the action a which has the highest value $Q(s, a)$. In contrast to this, an ε-greedy policy allows exploratory actions by choosing a random action with likelihood ε.

More exploration usually facilitates faster learning, so a value of $\varepsilon = 0.6$ was used in our learning experiments. The learning rate α was set to a fixed value of 0.001, the discounting factor γ was set to 0.9, which are standard values and led to good results in our experiments.

This technique results in a policy that tells the robot when to select the feature-based representation and when to choose the grid map. Note that our approach to learn a strategy for making decisions is independent of the actual feature detector used. One could even use this approach to choose among multiple feature detectors.

The overall mapping algorithm is depicted in Algorithm 2.

Algorithm 2 Our combined approach

Require:
$\mathcal{S}_{t-1}$, the sample set of the previous time step
$z_{l,t}$, the most recent laser scan
$z_{f,t}$, the most recent feature measurement
u_{t-1}, the most recent odometry measurement
Ensure:
$\mathcal{S}_t$, the new sample set

maptype = decide($\mathcal{S}_{t-1}, z_{l,t}, z_{f,t}, u_{t-1}$)

$\mathcal{S}_t = \{\}$
for all $s_{t-1}^{(i)} \in \mathcal{S}_{t-1}$ **do**
$< x_{t-1}^{(i)}, w_{g,t-1}^{(i)}, w_{f,t-1}^{(i)} m_{g,t-1}^{(i)}, m_{f,t-1}^{(i)} >= s_{t-1}^{(i)}$

// compute proposal
if (**maptype** = **grid**) **then**
$x_t^{(i)} \sim P(x_t \mid x_{t-1}^{(i)}, u_{t-1}, z_{l,t})$
else
$x_t^{(i)} \sim P(x_t \mid x_{t-1}^{(i)}, u_{t-1})$
end if

// update importance weights
$w_{g,t}^{(i)} = \text{updateGridWeight}(w_{g,t-1}^{(i)}, m_{g,t-1}^{(i)}, z_{l,t})$
$w_{f,t}^{(i)} = \text{updateFeatureWeight}(w_{f,t-1}^{(i)}, m_{f,t-1}^{(i)}, z_{f,t})$

// update maps
$m_{g,t}^{(i)} = \text{integrateScan}(m_{g,t-1}^{(i)}, x_t^{(i)}, z_{l,t})$
$m_{f,t}^{(i)} = \text{integrateFeatures}(m_{f,t-1}^{(i)}, x_t^{(i)}, z_{f,t})$

// update sample set
$\mathcal{S}_t = \mathcal{S}_t \cup \{< x_t^{(i)}, w_{g,t}^{(i)}, w_{f,t}^{(i)}, m_{g,t}^{(i)}, m_{f,t}^{(i)} >\}$
end for

for $i = 1$ to N **do**
if (**maptype** = **grid**) **then**
$w^{(i)} = w_g^{(i)}$
else
$w^{(i)} = w_f^{(i)}$
end if
end for

$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} (w^{(i)})^2}$
if $N_{\text{eff}} < T$ **then**
$\mathcal{S}_t = \text{resample}(\mathcal{S}_t, \{w^{(i)}\})$
end if

6. Experiments

Our approach has been evaluated using simulated and real robot data. The experiments have been designed to verify that our mapping approach is able to reduce the error compared to the purely feature-based technique (FastSLAM [13]) and to the purely grid-based approach [16]. We specifically considered environments which cannot be mapped accurately using one single model. In those cases the result is the same as using the original approach.

6.1. Simulation experiments

For generating the simulated data, we used the Carnegie Mellon Robot Navigation Toolkit. The simulated environment used to test our approach is shown in Fig. 2. It shows two symmetric buildings connected by an alley. The environment is spanning 70 m in total. We simulated a laser range finder with a maximum range of 4 m which is less than the distance between the trees in the alley (5 m). This limited sensor range is a realistic setting since it corresponds to the maximum range of small scale laser scanners such as the Hokuyo URG.

The motivating example in the introduction of this paper shows example results obtained with the different approaches. Fig. 1(a) is the result of the purely feature-based FastSLAM approach. Since no features are found inside the building structures, the robot cannot correct its trajectory inside the buildings. In contrast, the trajectory through the alley is well approximated using this approach.

The purely grid-based approach [16] is able to correctly map the buildings but introduces large errors in the alley (see Fig. 1(b)). Due to the limited range of the sensor, too few obstacles are observed and therefore no accurate scan registration is possible and thus the grid-based approach fails to map the alley appropriately.

In contrast to this, our combined approach using the learned policy is able to correct the trajectory of the robot all the time by selecting the appropriate model. It uses the grid maps inside the buildings and the features outside. The resulting map is shown in Fig. 1(c).

To evaluate our approach more quantitatively, we repeated this experiment 20 times with different random seeds. We compared our approach to the pure feature-based approach and the pure grid-based approach. The results in Fig. 3 show, that the combined approach is significantly better than both pure approaches (0.05 significance).

In addition to this, we compared the solution obtained by SARSA with those of the scan-matching heuristic and the N_{eff} heuristic described above. We measured the absolute deviation from ground truth in every time step. Fig. 4 illustrates that the average error of the learned model selection policy is lower than when using the heuristics. However, we could not show that this improvement is significant.

One interesting fact can be observed when comparing the results of these three technique by manual inspection. Even if the error measured as the deviation from the ground truth is not significantly smaller for the learned policy, the maps typically look nicer. The scan-match heuristic for example relies on a fixed threshold c_1. If the threshold is not optimally tuned, it can happen that the grid approach is not selected even though it would be the better model.

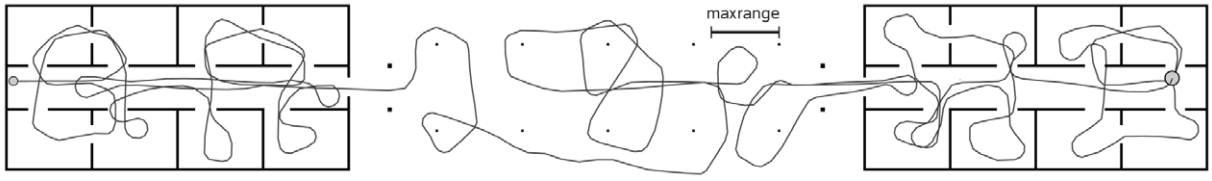


Fig. 2. Simulated environment used to test our approach. Shown are the ground truth map and trajectory of the robot.

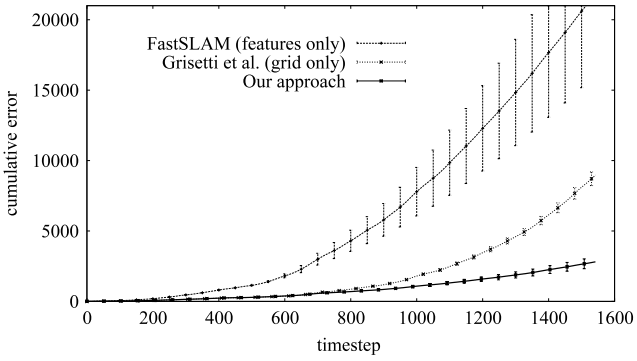


Fig. 3. Deviation of the weighted mean of the samples from ground truth using grid- and feature-model on their own and using the combined approach. The error bars illustrate the 0.05 confidence level.

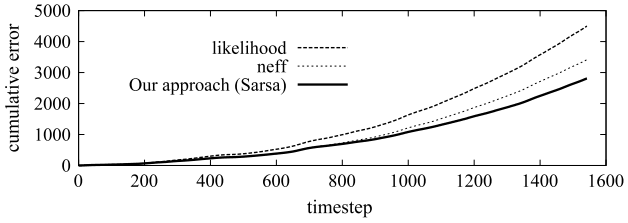


Fig. 4. Deviation of the weighted mean of the samples from ground truth using the scan-match likelihood heuristic, the N_{eff} heuristic and our approach.

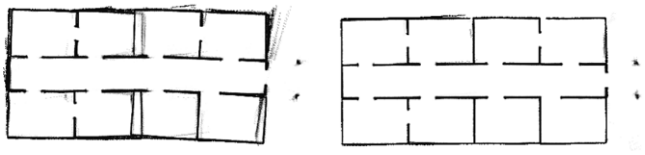


Fig. 5. Typical mapping results when using the likelihood heuristic (left) and our SARSA-based approach (right).

This leads to walls which are blurred or slightly misaligned. Fig. 5 depicts a magnified view of two maps illustrating the difference. Unfortunately, it is hard to design a measure that is able to take this blurriness into account. A similar effect can be observed when using the N_{eff} criterion.

Fig. 6 shows the decisions our algorithm made while processing the simulated dataset. One can clearly see that the grid map is used for pose estimation inside the buildings while the feature map is used outside of the buildings. At a first glance it looks as if the system used the wrong model around time-step 1000. Using features here is correct though since the robot entered the building to the right only briefly and then moved in the outdoor part again until approximately time-step 1100.

6.2. Real world experiments

Two real world data sets used in this experiment have been recorded at Freiburg University. The experiments have been conducted using an ActivMedia Pioneer 2-AT robot equipped with a SICK LMS laser range finder.

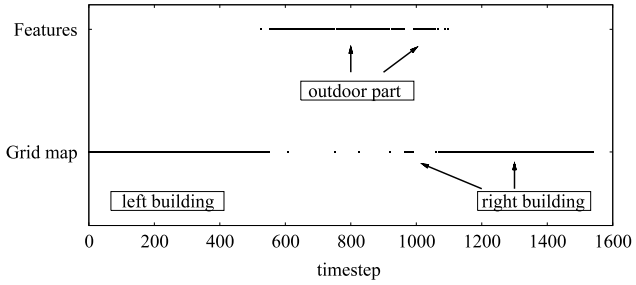


Fig. 6. Active representation chosen by our learned approach.

Fig. 7. Test environment with poles.

6.2.1. Poles

This experiment has been conducted in an office-building and on the street in front of the building. Since the outdoor environment does not contain a sufficient amount of detectable features, 20 artificial landmarks (poles) have been placed there. We used poles with a diameter of 15 cm and a height of about 100 cm (see Fig. 7). The robot was manually steered through the environment. It started outdoors in front of the building, went through the landmarks and then entered the building. After traversing the building the robots returns to the outdoor area and finishes its trajectory next to the starting location. To prevent the laser-scanner from detecting neighboring buildings, the sensor-range has been limited to 5 m. Again, this maximum range is not artificially bad but corresponds to small scale laser range scanners.

Since no ground truth was available, we measured the error against an approximated robot path which was generated using the grid-based approach of Grisetti et al. [16] with the full sensor range of the SICK laser scanner. Due to the 80 m sensing range, the robot always observed enough obstacles to build an accurate map. The resulting map and the obtained trajectory can be seen in Fig. 8.

We compared the results from our approach to those generated by a pure grid- and feature-based approach. Looking at the exemplary results in Fig. 9 notable differences in the quality of the maps can be seen. While the grid-based approach performs very well inside of the building it introduces numerous false matches in the

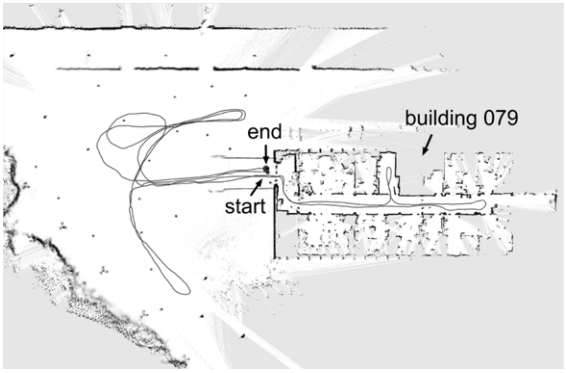


Fig. 8. Grid map of environment and approximated robot trajectory.

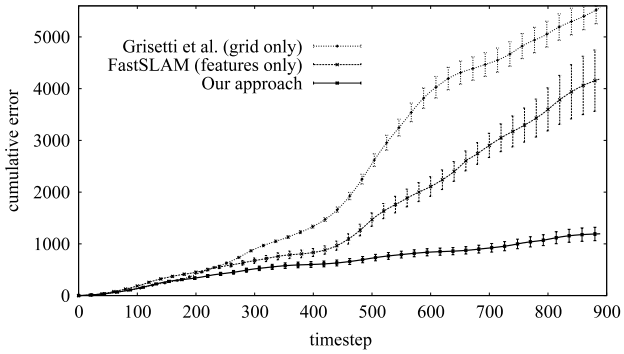


Fig. 10. Results of the poles experiment. The cumulative error in the pose estimation measured against the approximated ground truth trajectory. The error bars correspond to the 0.05 confidence level.

outdoor area. In contrast, the feature based approach is able to map the outdoor part well but is obviously not suited to correct odometry errors inside the building. Combining the strengths of both approaches, our combined method leads to an overall consistent map.

To evaluate our approach quantitatively we repeated the mapping process for 20 times. Fig. 10 plots the cumulative deviation from the approximated ground truth trajectory for each of the three evaluated strategies. The results confirm the results of the simulated experiment. They show that the combined approach performs significantly better than both pure approaches (0.05 significance).

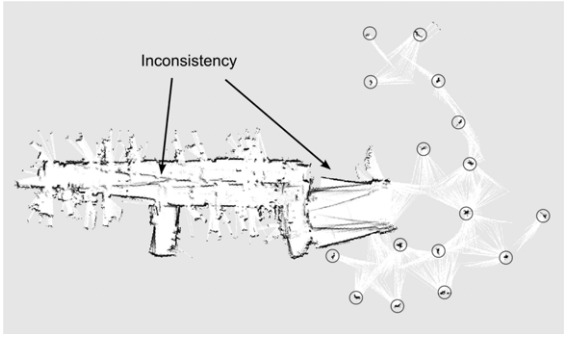
6.2.2. Parking lot

The Freiburg Computer Science campus includes a parking lot of about 50 m by 120 m (see Fig. 11). Lamps are set in two rows at a distance of 16 m in one direction and 25 m in the other direction. The second dataset was recorded on this parking lot at a time when no cars were present and therefore only the lamps caused reflections of the laser beams. The robot was steered manually through a building, around the neighboring parking lot, and back into the building again. The trajectory is plotted in Fig. 12. To evaluate our approach, again we limited the maximum laser range of the scanner to a range which is considerably smaller than the distance between two lamps.

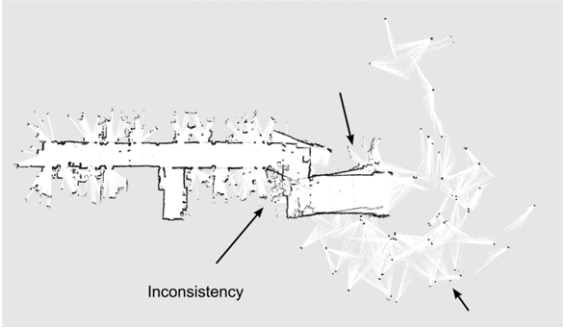
The approximated ground truth trajectory has been generated in the same way as we did in the first experiment. Fig. 13 shows the error of the weighted mean trajectory over time.

In summary, both real robot experiments lead to similar results as the experiment using simulated data. The combined approach performed significantly better compared to both traditional SLAM techniques using the limited sensor range.

The computational requirements of the presented approach are approximatively the sum of the individual techniques. On a notebook computer, our implementation runs online.



(a) Feature-based mapping system.



(b) Grid-based mapping system.

(c) Combined SLAM system using features and grid maps.

Fig. 9. Examples of resulting maps in the poles experiment. Using only a feature map (a) or a grid map (b) leads to inconsistent maps in this environment. Combining both representation yields a consistent map (c).

Fig. 11. Parking lot at Freiburg campus.

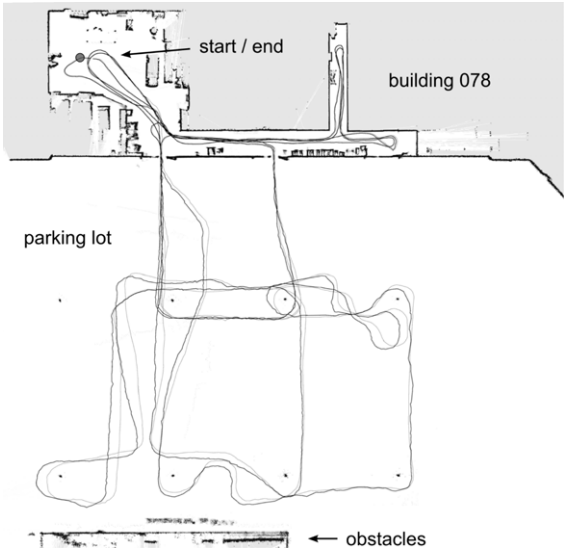


Fig. 12. Grid map of the parking lot and neighboring building 078 at the Freiburg campus. The approximated robot trajectory is shown in dark gray, the result of our combined mapping approach is shown in light gray.

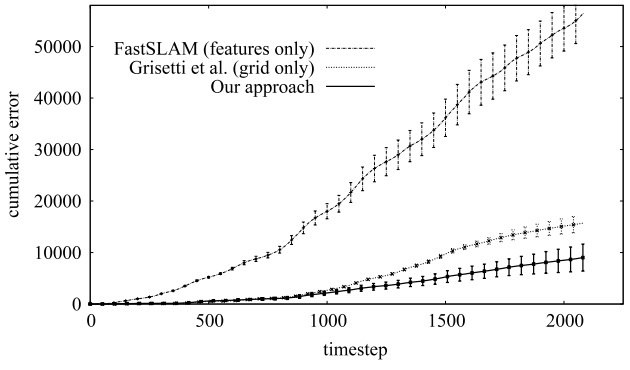


Fig. 13. Results of the parking lot experiment. Deviation of the weighted mean of the samples from the estimated trajectory (using the 80 m range scanner).

7. Conclusions

In this paper, we presented an improved approach to learning models of the environment with a Rao–Blackwellized particle filter. Our approach maintains feature maps as well as grid maps simultaneously to represent spatial structures. This allows the robot to select the model which provides the best expected estimates online. The model selection procedure is obtained by a reinforcement learning approach. The robot considers the previous estimate as well as the current observations to chose the model that will be used in the upcoming correction step. The process itself is independent of the actual feature detector. Our approach has been implemented and evaluated on real robot data as well as in simulation experiments. We showed that the presented technique allows a robot to more robustly learn maps of different types of environments. It outperforms traditional approaches that use only features or only grid maps. In real world experiments, we also showed that our approach is able to map environments which could not be modeled by either of the single approaches.

Acknowledgments

This work has partly been supported by the German Research Foundation (DFG) under contract number SFB/TR-8 (A3), and by the EC under contract number FP6-IST-034120-muFly.

References

[1] H.P. Moravec, A.E. Elfes, High resolution maps from wide angle sonar, in: Proc. of the IEEE Int. Conf. on Robotics & Automation, ICRA, St. Louis, MO, USA 1985, pp. 116–121.
[2] J.J. Leonard, H.F. Durrant-Whyte, Mobile robot localization by tracking geometric beacons, IEEE Transactions on Robotics and Automation 7 (4) (1991) 376–382.
[3] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial realtionships in robotics, in: I. Cox, G. Wilfong (Eds.), Autonomous Robot Vehicles, Springer Verlag, 1990, pp. 167–193.
[4] J. Neira, J.D. Tardós, Data association in stochastic mapping using the joint compatibility test, IEEE Transactions on Robotics and Automation 17 (6) (2001) 890–897.
[5] U. Frese, G. Hirzinger, Simultaneous localization and mapping—A discussion, in: Proc. of the IJCAI Workshop on Reasoning with Uncertainty in Robotics, Seattle, WA, USA, 2001, pp. 17–26.
[6] S. Julier, J. Uhlmann, H.F. Durrant-Whyte, A new approach for filtering nonlinear systems, in: Proc. of the American Control Conference, Seattle, WA, USA, 1995, pp. 1628–1632.
[7] J. Uhlmann, Dynamic Map Building and Localization: New Theoretical Foundations, Ph.D. Thesis, University of Oxford, 1995.
[8] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, H.F. Durrant-Whyte, Simultaneous localization and mapping with sparse extended information filters, Journal of Robotics Research 23 (7/8) (2004) 693–716.
[9] R. Eustice, H. Singh, J.J. Leonard, Exactly sparse delayed-state filters, in: Proc. of the IEEE Int. Conf. on Robotics & Automation, ICRA, Barcelona, Spain, 2005, pp. 2428–2435.
[10] A. Doucet, J.F.G. de Freitas, K. Murphy, S. Russel, Rao–Blackwellized partcile filtering for dynamic bayesian networks, in: Proc. of the Conf. on Uncertainty in Artificial Intelligence, UAI, Stanford, CA, USA, 2000, pp. 176–183.
[11] K. Murphy, Bayesian map learning in dynamic environments, in: Proc. of the Conf. on Neural Information Processing Systems, NIPS, Denver, CO, USA, 1999, pp. 1015–1021.
[12] M. Montemerlo, S. Thrum, D. Koller, B. Wegbreit, FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, in: Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI, 2003, pp. 1151–1156.
[13] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A factored solution to simultaneous localization and mapping, in: Proc. of the National Conference on Artificial Intelligence, AAAI, Edmonton, Canada, 2002, pp. 593–598.
[14] A. Eliazar, R. Parr, DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks, in: Proc. of the Int. Joint Conf. on Artificial Intelligence, IJCAI, Acapulco, Mexico, 2003, pp. 1135–1142.
[15] D. Hähnel, W. Burgard, D. Fox, S. Thrun, An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS, 2003, pp. 206–211.
[16] G. Grisetti, C. Stachniss, W. Burgard, Improved techniques for grid mapping with Rao–Blackwellized particle filters, IEEE Transactions on Robotics 23 (1) (2007) 34–46.
[17] C. Stachniss, G. Grisetti, W. Burgard, N. Roy, Evaluation of gaussian proposal distributions for mapping with Rao–Blackwellized particle filters, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, San Diego, CA, USA, 2007.
[18] G. Grisetti, G.D. Tipaldi, C. Stachniss, W. Burgard, D. Nardi, Fast and accurate slam with Rao–Blackwellized particle filters, Journal of Robotics & Autonomous Systems 55 (1) (2007) 30–38.
[19] E.M. Nebot, J.I. Nieto, J.E. Guivant, The hybrid metric maps (HYMMs): A novel map representation for denseslam, in: Proc. of the IEEE Int. Conf. on Robotics & Automation, ICRA, 2004.
[20] R. Sim, J.J. Little, Autonomous vision-based exploration and mapping using hybrid maps and Rao–Blackwellised particle filters, Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, October 2006, pp. 2082–2089.

[21] A.A. Makarenko, S.B. Williams, F. Bourgoult, H.F. Durrant-Whyte, An experiment in integrated exploration, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS, Lausanne, Switzerland, 2002.
[22] K.L. Ho, P.M. Newman, Loop closure detection in slam by combining visual and spatial appearance, Robotics and Autonomous Systems 54 (9) (2006) 740–749.
[23] A. Doucet, On sequential simulation-based methods for bayesian filtering, Technical report, Signal Processing Group, Dept. of Engineering, University of Cambridge, 1998.
[24] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, 2005, pp. 171–172.
[25] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.

Kai M. Wurm is a research scientist at the University of Freiburg (Germany). He studied computer science at the University of Freiburg and received his diploma degree in 2007. His research interests lie in the fields of SLAM, multi-robot exploration, and terrain classification.

Cyrill Stachniss studied computer science at the University of Freiburg and received his Ph.D. degree in 2006. After his Ph.D., he was a senior researcher at ETH Zurich. Since 2007, he has been an academic advisor at the University of Freiburg in the Laboratory for Autonomous Intelligent Systems. His research interests lie in the areas of robot navigation, exploration, SLAM, as well as learning approaches.

Giorgio Grisetti is working as a Post-doc at the Autonomous Intelligent Systems Lab of the University of Freiburg. Up to 2006, he was a Ph.D. student at University of Rome "La Sapienzia" in the Intelligent Systems Lab. His advisor was Daniele Nardi and he received his Ph.D. degree in April 2006. His research interests lie in the areas of mobile robotics. His previous and current work aims to provide effective solutions to mobile robot navigation in all its aspects: SLAM, localization, and path planning.